What is claimed is:

1. A method of processing fragments of a 3D image, the method comprising:

positioning a depth filter on a z-axis in a 3D space;

first comparing a depth value of each of a plurality of fragments forming a first object being rasterized in the 3D space with a depth value of the depth filter;

storing a first object data corresponding to each of the fragments of the first object and the first comparison data in a storage device;

rendering the fragments of the first object and second comparing a depth value of each of a plurality of fragments forming a second object being rasterized in the 3D space with the depth value of the depth filter;

removing a fragment from the fragments of the second object that overlap a fragment of the first object using the first object data stored in the storage device and the second comparison data; and

rendering the fragments of the first object and the fragments of the second object.

2. The method of claim 1, wherein the storage device has a $\log_2(n+1)$-bit storage space for each fragment, where n is a natural number.

3. The method of claim 1, wherein in the depth filter positioning step, the depth filter is positioned in a predetermined position on the z-axis.

4. A method of sequentially rendering a first object and a second object in a 3D image, the method comprising:

positioning a depth filter on a z-axis in a 3D space;

first comparing a depth value of each of a plurality of fragments of the first object being rasterized in the 3D space with a depth value of the depth filter;

storing first object data, which is mapped to the depth filter, and corresponds to each of the fragments of the first object and the first comparison data in a storage device;

rasterizing each of the fragments of the second object;

second comparing a depth value of each of the fragments of the second object with the depth value of the depth filter;

removing a fragment from the second object, using the second comparison data and the first comparison data that is stored in the storage device; and

outputting a fragment data that is formed of the first object data and the second object data.

5. The method of claim 4, wherein in the depth filter positioning step, when n depth filters are positioned on the z-axis, the storage device is implemented as a SRAM.

6. The method of claim 5, wherein the storage device has a $\log_2 (n+1)$-bit storage space for each fragment, where n is a natural number.

7. The method of claim 4, wherein in the depth filter positioning step, when n depth filters are positioned on the z-axis, the storage device includes a cache memory and an external memory device, in which the cache memory is implemented as a SRAM and the external memory device is implemented as a SDRAM.

8. The method of claim 7, wherein at least one of the cache memory and the external memory device has a $\log_2 (n+1)$-bit storage space for each fragment.

9. The method of claim 4, wherein in the depth filter positioning step, n depth filters are positioned in a predetermined number of different positions on the z-axis.

10. The method of claim 4, further comprising:

receiving the fragment data;

performing a texturing;

outputting the result of the texturing;

receiving the result of the texturing;

performing a per-fragmenting

outputting the result of the per-fragmenting; and

receiving the result of the per-fragmenting and a depth value of an image being displayed;

third comparing the depth value of each of the fragments of the first object with the depth value of each of the fragments of the second object, based on the result of the per-fragmenting and the depth value of the image; and

removing a fragment from the fragments of the second object, based on the third comparison data.

11. The method of claim 10, further comprising:

fourth comparing the fragments of the second object that cannot be removed in the fragment removal step with the fragments of the second object that can be removed but have not been removed; and

controlling a position of the depth filter based on the fourth comparison data.

12. A 3D graphics rendering engine comprising:

an internal memory device, which stores data;

a pixel interpolating circuit, which receives 3D information and creates data for the pixels within a triangle, based on the received 3D information;

a texel interpolating circuit, which creates data for the coordinates within the triangle in response to an output signal of the pixel interpolating circuit; and

a depth filtering circuit, which includes n depth filters positioned on a z-axis in a 3D space, wherein the depth filtering circuit:

first compares a depth value of each of a plurality of fragments forming a first object being rasterized in a 3D space with a depth value of each of the n depth filters;

stores first object data, which is mapped to the depth filter and corresponds to each of the fragments of the first object and the first comparison data in a storage device;

rasterizes each of a plurality of fragments forming a second object,;

second compares a depth value of each of the fragments of the second object with the depth value of each of the n depth filters;

removes a fragment from the fragments of the second object that overlap a fragment of the first object using the first object data stored in the storage device and the second comparison data; and

outputs a fragment data formed of the first object data and the second object data.

13. The 3D graphics rendering engine of claim 12, wherein the storage device has a $\log_2 (n+1)$-bit storage space for each fragment, where n is a natural number.

14. The 3D graphics rendering engine of claim 12, wherein the n depth filters are positioned in different positions on the z-axis.

15. The 3D graphics rendering engine of claim 12, further comprising:

a texture block, which receives the fragment data, performs a texturing, and outputs the result of the texturing;

a per-fragment block, which receives the result of the texturing, performs a per-fragmenting, and outputs the result of the per-fragmenting; and

a depth test block, which receives the result of the per-fragmenting and a depth value of an image being displayed, wherein the depth test block:

third compares the depth value of each of the fragments of the first object with the depth value of each of the fragments of the second object, based on the received result of the per-fragmenting and the received depth value of the image; and

removes a fragment from the fragments of the second object, based on the third comparison data.

16. The 3D graphics rendering engine of claim 12, wherein the depth test block:

fourth compares the fragments of the second object that cannot be removed through the fragment removal step with the number of fragments of the second object that can be removed but have not been removed; and

outputs a position control signal for controlling a position of the depth filter based on the fourth comparison data.